AWS provides numerous services and in the beginning it is difficult to find out where to get started. An easy way to start experimenting with AWS, is to make use of [AWS Elastic Beanstalk](https://aws.amazon.com/elasticbeanstalk/). Elastic Beanstalk can be seen as an abstraction layer above core AWS services (like Amazon EC2, Amazon Elastic Container Service (Amazon ECS), Auto Scaling, and Elastic Load Balancing). It provisions and operates the infrastructure and manages the application stack for you, in order for you to focus on writing code.

 Create a Spring Boot App

Now that we have setup our AWS account, we need to create a simple Spring Boot App. We will create a Spring Web MVC application with a HelloController which just returns a hello message including the IP address.

We go to [Spring Initializr](https://start.spring.io/), select the Spring Web dependency, use Spring Boot 2.3.4, Java 11 and generate the project. The HelloController looks as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | @RestController  public class HelloController {        @GetMapping("/hello")      public String hello() {          String message = "Hello AWS Elastic Beanstalk!";          try {              InetAddress ip = InetAddress.getLocalHost();              message += " From host: " + ip;          } catch (UnknownHostException e) {              e.printStackTrace();          }          return message;      }    } |

Run the application:

|  |  |
| --- | --- |
| 1 | $ mvn spring-boot:run |

And invoke the URL:

|  |  |
| --- | --- |
| 1  2 | $ curl http://localhost:8080/hello  Hello AWS Elastic Beanstalk! From host: your-computer-name/127.0.1.1 |

The sources of this project can be found at [GitHub](https://github.com/mydeveloperplanet/MyElasticBeanstalkPlanet).

5. Install and Configure EB CLI

Before we can start with deploying our application, we need to install and configure the Elastic Beanstalk CLI tool which will allow us to deploy.

5.1 Install EB CLI

The instructions for installing the EB CLI can be found at [GitHub](https://github.com/aws/aws-elastic-beanstalk-cli-setup). We describe the instructions for Ubuntu 20.04.

First, we clone the git repository:

|  |  |
| --- | --- |
| 1 | $ git clone https://github.com/aws/aws-elastic-beanstalk-cli-setup.git |

Next, run the installer from the directory where you executed the git clone command:

Then we run the installer again:

|  |  |
| --- | --- |
|  | $ ./aws-elastic-beanstalk-cli-setup/scripts/bundled\_installer |

This did the trick, the EB CLI is successfully installed!

Last thing to do is to ensure that the eb command is available in your path.

|  |  |
| --- | --- |
| 1 | $ echo 'export PATH="/home/gunter/.ebcli-virtual-env/executables:$PATH"' >> ~/.bash\_profile && source ~/.bash\_profile |

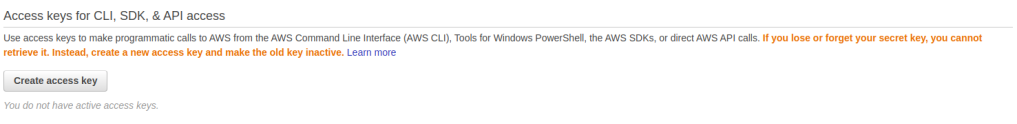
5.2 Configure the EB CLI

Next step is to configure the EB CLI. Navigate to your project directory and run the eb initialisation command:

|  |  |
| --- | --- |
|  | $ eb init  Select a default region  1) us-east-1 : US East (N. Virginia)  2) us-west-1 : US West (N. California)  3) us-west-2 : US West (Oregon)  4) eu-west-1 : EU (Ireland)  5) eu-central-1 : EU (Frankfurt)  6) ap-south-1 : Asia Pacific (Mumbai)  7) ap-southeast-1 : Asia Pacific (Singapore)  8) ap-southeast-2 : Asia Pacific (Sydney)  9) ap-northeast-1 : Asia Pacific (Tokyo)  10) ap-northeast-2 : Asia Pacific (Seoul)  11) sa-east-1 : South America (Sao Paulo)  12) cn-north-1 : China (Beijing)  13) cn-northwest-1 : China (Ningxia)  14) us-east-2 : US East (Ohio)  15) ca-central-1 : Canada (Central)  16) eu-west-2 : EU (London)  17) eu-west-3 : EU (Paris)  18) eu-north-1 : EU (Stockholm)  19) eu-south-1 : EU (Milano)  20) ap-east-1 : Asia Pacific (Hong Kong)  21) me-south-1 : Middle East (Bahrain)  22) af-south-1 : Africa (Cape Town)  (default is 3): 17 |

We choose 17 because that is the area we are residing.

In the next step, we need a Security Access key. Go to the AWS Management Console, click My Account – My Security Credentials. We need to create an access key.



Click the Create access key button. In a popup window your Access key ID and Secret access key are shown. This information will be shown only once, so make sure you store it somewhere in a safe place.

Now fill in the Access Key and Secret and continue:

|  |  |
| --- | --- |
| 1  2  3  4 | You have not yet set up your credentials or your credentials are incorrect  You must provide your credentials.  (aws-access-id): <fill in your access id>  (aws-secret-key): <fill in your secret key> |

Choose an application name,we keep the default and choose Java as platform because we created a Spring Boot jar file.

Last two questions are answered with no. CodeCommit will store your code in AWS CodeCommit, this will speed up deployments, but for our small example, this is not necessary. The SSH keys are needed when you want to have acces via SSH.

|  |  |
| --- | --- |
| 1  2  3 | Do you wish to continue with CodeCommit? (Y/n): n  Do you want to set up SSH for your instances?  (Y/n): n |

In our repository, a config.yml file is added to directory .elasticbeanstalk with the following contents:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | branch-defaults:    master:      environment: null      group\_suffix: null  global:    application\_name: MyElasticBeanstalkPlanet    branch: null    default\_ec2\_keyname: null    default\_platform: Corretto 11 running on 64bit Amazon Linux 2    default\_region: eu-west-3    include\_git\_submodules: true    instance\_profile: null    platform\_name: null    platform\_version: null    profile: eb-cli    repository: null    sc: git    workspace\_type: Application |

The Elastic Beanstalk files are also automatically added to the .gitignore file.

The configuration of EB CLI is ready!

6. Deploy to AWS

A few things are left to do before we can deploy the Spring Boot App to Elastic Beanstalk. The Elastic Beanstalk environments run an nginx instance on port 80 to proxy the actual application, running on port 5000. Therefore, we need to set the server port to port 5000 in the applications.properties file.

|  |  |
| --- | --- |
| 1 | server.port=5000 |

Now build the application which will create the jar file target/MyElasticBeanstalkPlanet-0.0.1-SNAPSHOT.jar:

|  |  |
| --- | --- |
| 1 | $ mvn clean install |

Next, we need to add the following to the .elasticbeanstalk/config.yml:

|  |  |
| --- | --- |
| 1  2 | deploy:    artifact: target/MyElasticBeanstalkPlanet-0.0.1-SNAPSHOT.jar |

Finally, we are going to create our AWS environment. Do so with the -s option, otherwise a loadbalancer is created which will cost extra. The -s option will create a single instance. Wait a few minutes and the environment will be available.

Lets check whether we can access our application (the URL is taken from the last but one log line):

|  |  |
| --- | --- |
| 1  2 | $ curl http://MyElasticBeanstalkPlanet-dev.eu-west-3.elasticbeanstalk.com:80/hello  Hello AWS Elastic Beanstalk! From host: ip-172-31-47-35.eu-west-3.compute.internal/172.31.47.35 |

With the command eb console the console of your environment can be viewed. Just try it out and navigate the pages. Quite some information about your running application is available.

What if we want to change something and want to redeploy our application? This is quite simple. Let’s make a small change to the hello message and add the word again to the text to be displayed:

|  |  |
| --- | --- |
| 1 | String message = "Hello again AWS Elastic Beanstalk!"; |

Rebuild the application with mvn clean install and execute the eb deploy command in order to deploy the application:

Run the curl command again and the updated message is returned:

|  |  |
| --- | --- |
| 1  2 | $ curl http://MyElasticBeanstalkPlanet-dev.eu-west-3.elasticbeanstalk.com:80/hello  Hello again AWS Elastic Beanstalk! From host: ip-172-31-47-35.eu-west-3.compute.internal/172.31.47.35 |

7. Terminate the environment

At the end, it is wise to remove the environment. This can be done with the eb terminate command following the environment name. After a few minutes, the environment is gone.